

Integration of the JetFinder package in the ElectronPhoton analysis environment

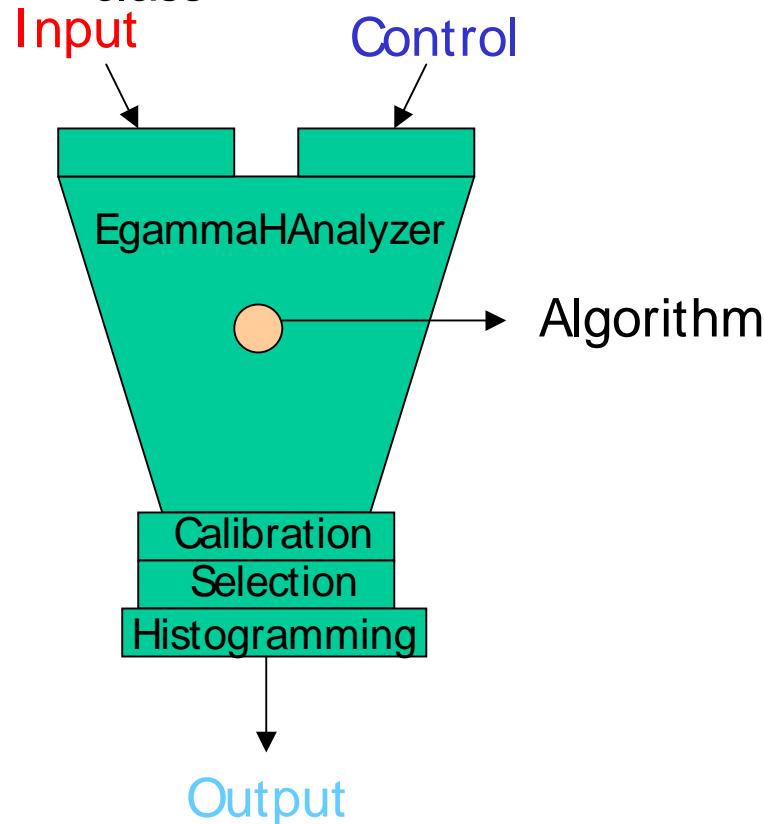
Physics Context: Study of combined electron-jet triggers

Software Context: ElectronPhoton & JetFinder Packages

Objectives: Being able to use the JetFinders in the ElectronPhoton analysis flow.

ElectronPhoton analysis flow

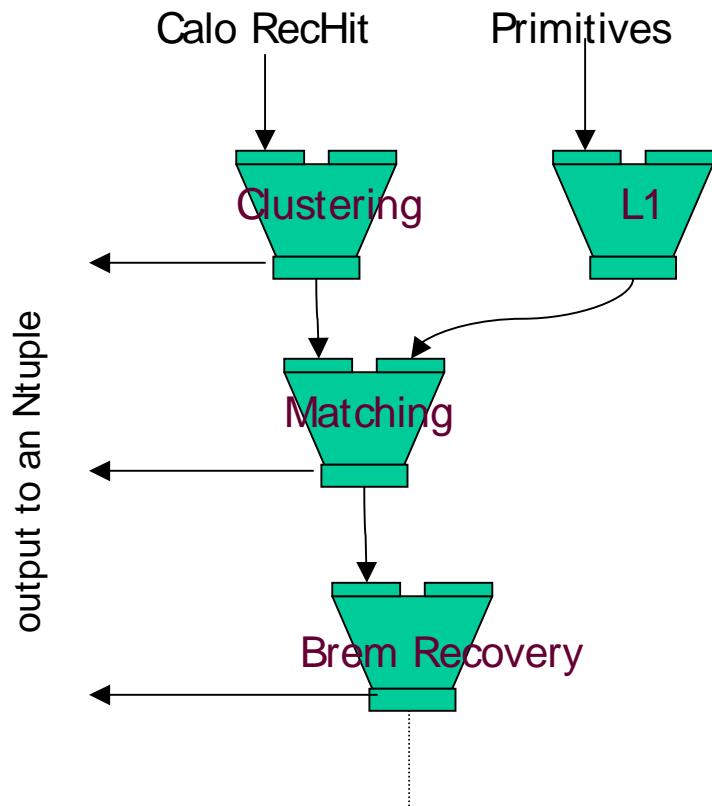
- Each Analysis step is represented by an: EgammaHAnalyzer template class



- The EgammaHAnalyzer specifies how it handles the Input and Control using the Algorithm to produce and output.
- It specifies how it applies the cuts/calibration on the produced objects.
- It is a RecUnit for the Output Objects.

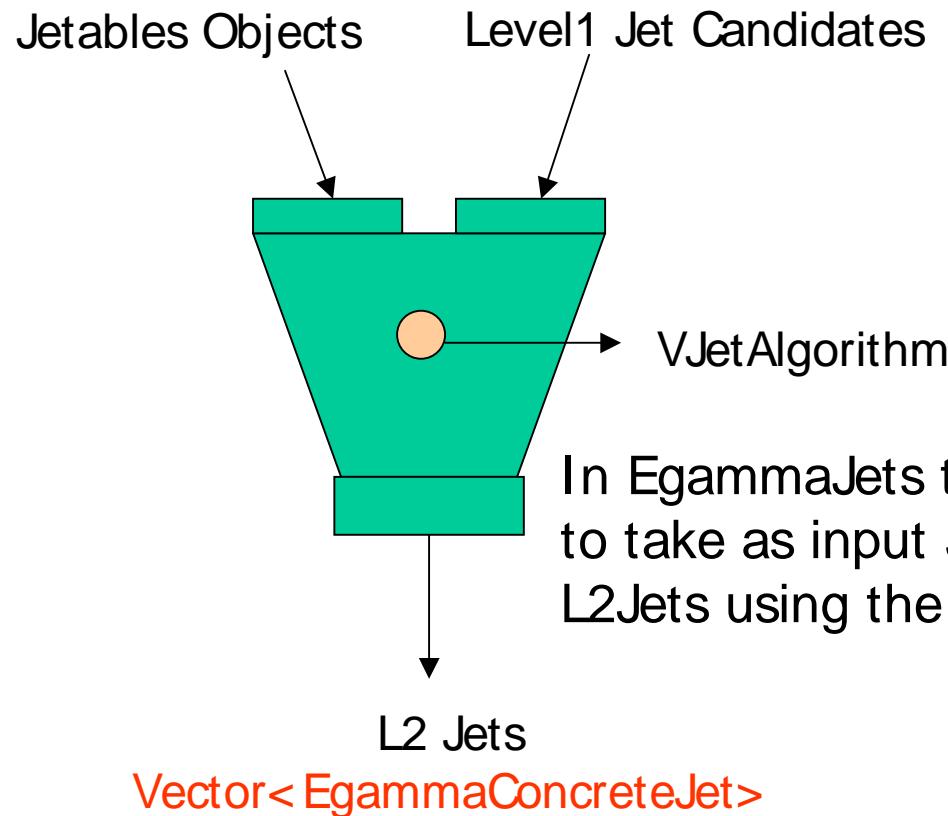
- EgammaHAnalyzer receive **input** Objects (ex: Cristals) uses **control** objects (ex: Clustering parameters) produces **output** objects (ex: clusters).

ElectronPhoton analysis flow: EgammaHAnalyzers can be chained



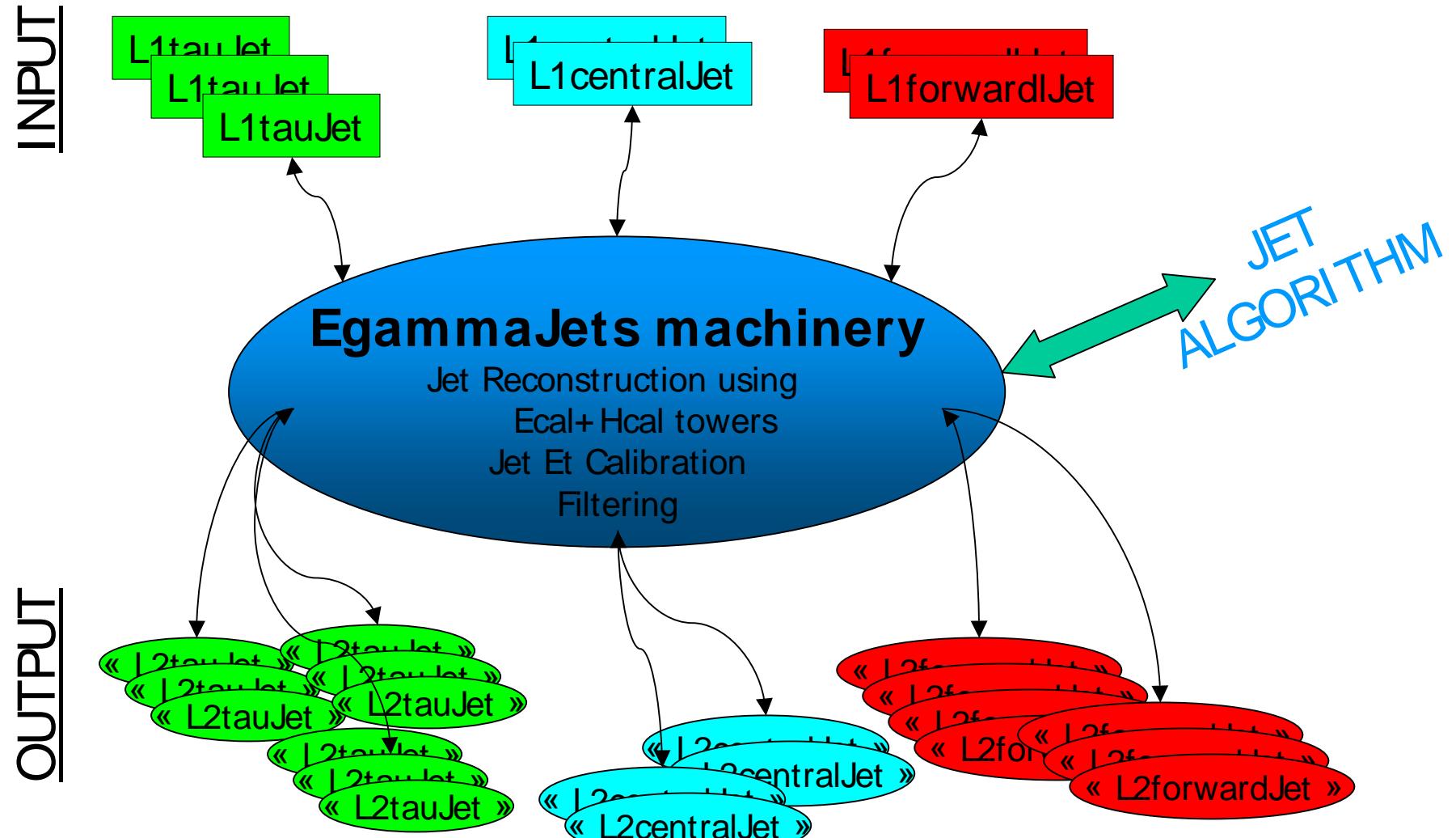
- At each Step we can choose to store the information of the intermediate objects.
- For each step, cuts and/or calibration can be applied.
- The analysis is defined by the module sequence and you can add more if you want to extend the analysis.

EgammaJets: Possibility to transparently use the JetFinders in the analysis flow



In EgammaJets the EgammaHAnalyzer is specialized to take as input JetableObjects, and to produce the L2Jets using the existing JetFinder package

What does it do ?



Calibration, Selection & Histogramming

- Calibration:

The Et of each reconstructed jet can be calibrated Using Calibration constants from the Jet-Met group.

It can be easily switched on or off:

```
JetAnalyzer->SetCalib(new ECJEnergyCorr);
```

- Selection:

The reconstructed jets can be selected on any criteria.

Ex: They can be sorted using the transverse energy

Ex: You can select to reconstruct jets only in L1tJet regions

```
JetAnalyzer->SetCut(new ECJSorting);
JetAnalyzer->AddCut(new ECJSelect("L1tJET"));
```

- Histogramming:

You can specify whether you want or not to histogram the Jets after calibration and selection.

How to use it ?

- In the constructor of your analysis class:

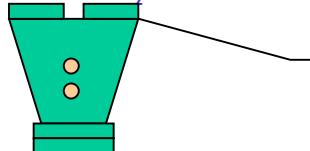
- ❖ Instances of the Algorithms and the generator of the JetableObjects

```
theICAlgo=new IterativeConeAlgorithm<ConcreteJet>;
theGenInput=new HLTJetFinderEcalPlusHcalTowerInput;
theJetFinder=new EgammaJetFinder(theICAlgo);
theProducer=new VJetInputProducer(theGenInput);

dummy=new InputProducer<Dummy>;
dummyalgo=new Level1DummyAlgo();
h=Singleton<EgammaHbook4Histogrammer>::instance();
```

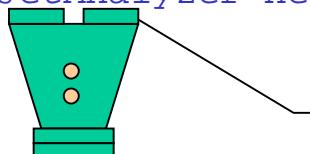
- ❖ Instances of the RecUnits for the L1Jets and the L2Jets

```
L1Analyzer=new EgammaHAnalyzer <Level1HObject,Dummy,Dummy>(dummyalgo,h,dummy);
```



RecUnit that produces **Level1HObject** containing all L1Jet candidates.

```
JetAnalyzer=new EgammaHAnalyzer <EgammaConcreteJet,Dummy,EgammaVJetableObjects>
(theJetFinder,h,theProducer,dummy);
```



RecUnit that produces **EgammaConcreteJet(s)**
Each EgammaConcreteJet contains (Et,Energy,Eta,Phi,Mass,L1 association)

How to use it ? (continued)

- ❖ Setting up calibration and filtering

```
JetAnalyzer->SetCalib(new ECJEnergyCorr);
JetAnalyzer->SetCut(new ECJSorting);
JetAnalyzer->AddCut(new ECJSelect("L1tJET"));
```

- in `update(G3EventProxy* ev)` ⇔ called for each event

If you need the Level1 candidates:

```
RecItr<Level1HObject> theL1(ev->RecEvent(),"Level1HObject");
If(theL1->next()) cout << theL1->nxJet('c') << endl;
```

Now to get all Jets constructed in each L1 jet region:

```
RecItr<EgammaConcreteJet> myJets(ev->RecEvent(),"EgammaConcreteJets");
while(myJets.next()) cout << *myJets << endl;
```

The info stored in the Ntuple

3 Blocks: L1, L2, Jet

- L1: contains (n, Et, Eta, Phi) for the three type of jets.
Access: `EtL1tJet(l1index)`
- L2: contains (n, Et, Eta, Phi) for the jets reconstructed using the JetFinder in the L1 Regions:
Access: `EtL2tJet(l2index, l1index)`
- Jet: contains all jets reconstructed in all L1 Regions but not grouped by regions.

*	1	* I*4 *	* [0,4]	* L2	* nL2cJET
*	2	* I*4 *	*	* L2	* nbL2cJET(nL2cJET)
*	3	* R*4 *	*	* L2	* EtL2cJET(10,nL2cJET)
*	4	* R*4 *	*	* L2	* EtaL2cJET(10,nL2cJET)
*	5	* R*4 *	*	* L2	* PhiL2cJET(10,nL2cJET)
*	6	* R*4 *	*	* L2	* MasL2cJET(10,nL2cJET)
*	7	* I*4 *	* [0,4]	* L2	* nL2fJET
*	8	* I*4 *	*	* L2	* nbL2fJET(nL2fJET)
*	9	* R*4 *	*	* L2	* EtL2fJET(10,nL2fJET)
*	10	* R*4 *	*	* L2	* EtaL2fJET(10,nL2fJET)
*	11	* R*4 *	*	* L2	* PhiL2fJET(10,nL2fJET)
*	12	* R*4 *	*	* L2	* MasL2fJET(10,nL2fJET)
*	13	* I*4 *	* [0,4]	* L2	* nL2tJET
*	14	* I*4 *	*	* L2	* nbL2tJET(nL2tJET)
*	15	* R*4 *	*	* L2	* EtL2tJET(10,nL2tJET)
*	16	* R*4 *	*	* L2	* EtaL2tJET(10,nL2tJET)
*	17	* R*4 *	*	* L2	* PhiL2tJET(10,nL2tJET)
*	18	* R*4 *	*	* L2	* MasL2tJET(10,nL2tJET)
*	1	* I*4 *	* [0,4]	* L1	* nL1cJET
*	2	* R*4 *	*	* L1	* L1cJETEt(nL1cJET)
*	3	* R*4 *	*	* L1	* L1cJETEta(nL1cJET)
*	4	* R*4 *	*	* L1	* L1cJETPhi(nL1cJET)
*	5	* I*4 *	* [0,4]	* L1	* nL1fJET
*	6	* R*4 *	*	* L1	* L1fJETEt(nL1fJET)
*	7	* R*4 *	*	* L1	* L1fJETEta(nL1fJET)
*	8	* R*4 *	*	* L1	* L1fJETPhi(nL1fJET)
*	9	* R*4 *	*	* L1	* L1TotEt
*	10	* R*4 *	*	* L1	* L1MEt
*	11	* I*4 *	* [0,4]	* L1	* nL1iEM
*	12	* R*4 *	*	* L1	* L1iEMEt(nL1iEM)
*	13	* R*4 *	*	* L1	* L1iEMEta(nL1iEM)
*	14	* R*4 *	*	* L1	* L1iEMPhi(nL1iEM)
*	15	* I*4 *	* [0,4]	* L1	* nL1nEM
*	16	* R*4 *	*	* L1	* L1nEMEt(nL1nEM)
*	17	* R*4 *	*	* L1	* L1nEMEta(nL1nEM)
*	18	* R*4 *	*	* L1	* L1nEMPhi(nL1nEM)
*	19	* I*4 *	* [0,4]	* L1	* nL1tJET
*	20	* R*4 *	*	* L1	* L1tJETEt(nL1tJET)
*	21	* R*4 *	*	* L1	* L1tJETEta(nL1tJET)
*	22	* R*4 *	*	* L1	* L1tJETPhi(nL1tJET)
*	23	* I*4 *	* [0,4]	* L1	* nL1uEM
*	24	* R*4 *	*	* L1	* L1uEMEt(nL1uEM)
*	25	* R*4 *	*	* L1	* L1uEMEta(nL1uEM)
*	26	* R*4 *	*	* L1	* L1uEMPhi(nL1uEM)

Conclusions & Prospects

- EgammaJets is an integration of existing jet algorithms with the Egamma analysis « framework »
 - Jets can be reconstructed in the L1Jet regions
 - Calibration using Jet-Met tables is included
 - Various cuts can be applied on the produced jet collection
 - Histogramming of the jet variables is automatic
-
- Make a full analysis for the $h \rightarrow \tau\tau \rightarrow e$ jet using this tool
 - Add backward tracking in this framework to validate tau-jet L2 candidates.
 - seeding, backward tracking exist
 - Include pixel info ?